Imperial to metric

```
import React, { useState, useEffect } from "react";
import {
  SafeAreaView,
  View,
  Text,
  TextInput,
  StyleSheet,
  ScrollView,
} from "react-native";
import { Picker } from "@react-native-picker/picker";

export default function App() {
  const [inputValue, setInputValue] = useState("");
  const [fromUnit, setFromUnit] = useState("cm");
  const [toUnit, setToUnit] = useState("inch");
  const [result, setResult] = useState("");

  // Factors relative to meter
  const unitFactors = {
    cm: 0.01,
    inch: 0.0254,
    feet: 0.3048,
    meter: 1,
    yard: 0.9144,
    mile: 1609.34,
    kilometer: 1000,
    micrometer: 1e-6,
    nanometer: 1e-9,
    millimeter: 0.001,
    nauticalMile: 1852,
    exameter: 1e18,
    petameter: 1e15,
    terameter: 1e12,
  };

  useEffect(() => {
```

```
  if (inputValue !== "" && !isNaN(inputValue)) {
    const valueInMeters = parseFloat(inputValue) * unitFactors[fromUnit];
    const converted = valueInMeters / unitFactors[toUnit];
    setResult(converted.toFixed(4));
  } else {
    setResult("");
  }
}, [inputValue, fromUnit, toUnit]);

return (
  <SafeAreaView style={styles.container}>
    <ScrollView contentContainerStyle={styles.scroll}>
      <Text style={styles.title}>Length Converter</Text>

      <TextInput
        style={styles.input}
        keyboardType="numeric"
        value={inputValue}
        onChangeText={setInputValue}
        placeholder="Enter value"
      />

      <View style={styles.pickerContainer}>
        <Picker
          selectedValue={fromUnit}
          onValueChange={(val) => setFromUnit(val)}
          style={styles.picker}
        >
          {Object.keys(unitFactors).map((unit) => (
            <Picker.Item key={unit} label={unit} value={unit} />
          ))}
        </Picker>
      </View>

      <View style={styles.pickerContainer}>
        <Picker
          selectedValue={toUnit}
```

```
          onValueChange={(val) => setToUnit(val)}
          style={styles.picker}
        >
          {Object.keys(unitFactors).map((unit) => (
            <Picker.Item key={unit} label={unit} value={unit} />
          ))}
        </Picker>
      </View>

      <Text style={styles.result}>
        {result !== "" ? `${inputValue} ${fromUnit} = ${result} ${toUnit}` : ""}
      </Text>
    </ScrollView>
  </SafeAreaView>
 );
}

const styles = StyleSheet.create({
 container: {
  flex: 1,
  backgroundColor: "#f8f8f8",
 },
 scroll: {
  flexGrow: 1,
  justifyContent: "center",
  alignItems: "center",
  padding: 20,
 },
 title: {
  fontSize: 26,
  fontWeight: "bold",
  marginBottom: 20,
 },
 input: {
  width: "100%",
  borderWidth: 1,
  borderColor: "#ccc",
```

```
    padding: 10,
    borderRadius: 8,
    marginBottom: 15,
    backgroundColor: "#fff",
  },
  pickerContainer: {
    width: "100%",
    borderWidth: 1,
    borderColor: "#ccc",
    borderRadius: 8,
    marginBottom: 15,
    backgroundColor: "#fff",
  },
  picker: {
    width: "100%",
  },
  result: {
    fontSize: 22,
    marginTop: 20,
    fontWeight: "bold",
  },
});
```